DRMS Processing Software
Reference Manual

Submitted To:
Guy Farley
Defence Research Establishment Ottawa
3701 Carling Ave.
Ottawa, Ontario
K1A 0Z4

Under Contract: W7714-8-0153
Atlantis Contract No.: 3112

May 28, 1999

ATLANTIS
Atlantis Scientific Inc.
20 Colonnade Road, Suite 110
Nepean, Ontario, Canada, K2E 7M6

105-03797/

Table of Contents

1 INTRODUCTION 1

2 RCS DATA PROCESSING SYSTEM 2

2.1 RAW DATA USED BY THE RCSDPS SYSTEM 2
2.2 CALCULATING THE RCS 3
2.2.1 RCS of Calibration Device *4*
2.2.2 Calculation of Range *4*
2.2.3 Calculation of Received Power *5*
2.3 CALCULATING THE RANGE RESOLUTION USING HRR DATA 7

3 RCSDPS SOFTWARE 8

3.1 VIEW RAW DATA 10
3.2 ANALYSE RCS BURST 14
3.3 CUSTOMIZE RCS CALIBRATION 19
3.4 PROCESS RCS RECORDING 22
3.5 COMPARE RCS RECORDING 34
3.6 PROCESS HRR RECORDING 40
3.7 SYSTEM CONFIGURATION 44

4 REFERENCES 45

5 APPENDIX A: SAMPLE CALIBRATION FILE 46

6 APPENDIX B: RCS FOR DREO CORNER REFLECTORS 48

7 APPENDIX C: ALGORITHMS 51

Table of Figures

Figure 2.1: Calculation of range to target. 5
Figure 3.1: Flow chart for RCSDPS main program. 9
Figure 3.2: Flow chart for program *viewraw*. 11
Figure 3.3: Flow chart for program *fileraw*. 12
Figure 3.4: Flow chart for program *getburst*. 13
Figure 3.5: Flow chart for program *brstanal*. 15
Figure 3.6: Flow chart for program *fileanal*. 16
Figure 3.7: Flow chart for program *validcal*. 17
Figure 3.8: Flow chart for program *getbrst2*. 18
Figure 3.9: Flow chart for program *avgmenu*. 18
Figure 3.10: Flow chart for program *cuscalib*. 20
Figure 3.11: Flow chart for program *filecal*. 21
Figure 3.12: Flow chart for program *process*. 24
Figure 3.13: Flow chart for program *fileproc*. 25
Figure 3.14: Flow chart for program *makercs*. 26
Figure 3.15: Flow chart for program *procparm*. 27
Figure 3.16: Flow chart for program *procrcs*. 28
Figure 3.17: Flow chart for program *procmenu*. 30
Figure 3.18: Flow chart for program *multparm*. 30
Figure 3.19: Flow chart for program *procsum1*. 31
Figure 3.20: Flow chart for program *procsum2*. 32
Figure 3.21: Flow chart for program *prcsmenu*. 33
Figure 3.22: Flow chart for program *compare*. 34
Figure 3.23: Flow chart for program *filecomp*. 35
Figure 3.24: Flow chart for program *comphis*. 36
Figure 3.25: Flow chart for program *compmenu*. 36
Figure 3.26: Flow chart for program *compsum1*. 37
Figure 3.27: Flow chart for program *crcsmenu*. 38
Figure 3.28: Flow chart for program *compsum2*. 38
Figure 3.29: Flow chart for program *comprcs*. 39
Figure 3.30: Flow chart for program *hrrproc*. 41
Figure 3.31: Flow chart for program *filehrr*. 42
Figure 3.32: Flow chart for program *gethrr*. 43
Figure 3.33: Flow chart for program *setifft*. 43
Figure 3.34: Flow chart for program *syscfg*. 44

1 Introduction

This document describes the details of the algorithms developed for the Radar Cross Section Data Processing System (RCSDPS). The structure of the RCSDPS software is also described in detail using flow charts to illustrate which programs and/or functions are executed for each of the options provided.

The RCSDPS software, written in MATLAB, is developed to process raw radar data obtained by the RCS Data Acquisition System (RCSDAS). Both the RCSDPS and the RCSDAS systems were developed by Atlantis Scientific Incorporated for the Defence Research Establishment, Ottawa.

2 RCS Data Processing System

The RCS Data Processing System (RCSDPS) has been developed to measure the radar cross section (RCS) of a given target using raw radar data obtained from the RCS Data Acquisition System (RCSDAS). It also has the capability to process high range resolution (HRR) data, obtained from the RCSDAS system, which is used to determine the relative locations of the major scatterers of the target. The data acquisition system and the raw data acquired by the system are described in detail in [1]. A brief description of the raw data used by the RCSDPS is given below. The rest of this section provides the details of how the RCS for a target is calculated and how the range resolution is resolved.

2.1 Raw Data Used by the RCSDPS System

Three different types of data, acquired using the RCSDAS system, are used by the RCSDPS system. These include

1. I and Q burst data,
2. HRR data and
3. Telemetry data.

The I and Q burst data is used to determine the RCS of a given target. This data contains one sample of I and Q data for every RF pulse transmitted in a given burst. A burst is defined as a consecutive group of pulses transmitted at a specified pulse repetition frequency (PRF). Several bursts are contained in one data set. The RCS can therefore be calculated on a pulse to pulse basis (within a single burst) or on a burst to burst basis. Included within the data are the RCSDAS system configuration parameters, such as the PRF and the transmit frequency range of the bursts, as well as the parameters for each burst. The burst parameters consist of the frequency, power, pulse width, automatic gain control (AGC), attenuation, range, transmit and receive polarizations, and the total number of pulses in the burst. An example of I and Q burst data is given in Appendix A.

High range resolution (HRR) data is similar to the I and Q burst data. The difference is that the synthesizer sweeps over a range of frequencies as the pulses are transmitted. One pulse is transmitted for each frequency in the sweep range. This data is not used to calculate the RCS, but is used to determine a high range resolution profile for the given target.

The telemetry data, if it exists, contains the time and bearing of a ship target. It is used for determining how the RCS of a ship changes as its bearing changes, and is not necessary for RCS calculations.

2.2 Calculating the RCS

The RCS Data Processing System (RCSDPS) is used to measure the radar cross section (RCS) of a given target by comparing the measured backscatter return from the target to the return of a known calibrated scatterer. The algorithm developed to measure the RCS is based on the radar range equation, which is given by

$$P_r = \frac{P_t G_t}{4\pi R^2} \frac{\sigma}{4\pi R^2} \frac{G_r \lambda^2}{4\pi} \frac{1}{L} \qquad (2.1)$$

where

σ is the RCS of the target,
P_t is the transmitted power,
G_t is the gain of the transmitting antenna,
G_r is the gain of the receiving antenna,
R is the range to the target,
L is a numerical factor to account for all losses.

By rearranging the above equation, the RCS of the target can then be calculated using the following equation

$$\sigma = \frac{P_r (4\pi)^3 R^4 L}{P_t G_t G_R \lambda^2}. \qquad (2.2)$$

Although the above equation can be used directly to measure the RCS, if all the parameters are known, a comparative measurement with a target of a known RCS is used instead. This is accomplished by assuming that the transmitted power, the wavelengths and the antenna gains are the same for both the target and the calibration device. Then the ratio of the RCS for the target to the RCS for the calibration device can be written as

$$\frac{\sigma_{tar}}{\sigma_{cal}} = \frac{(P_r)_{tar}}{(P_r)_{cal}} \frac{R_{tar}^4}{R_{cal}^4} \frac{L_{tar}}{L_{cal}}. \qquad (2.3)$$

The loss factors in equation 2.3 consist of all system losses. These include transmission losses, polarization losses and atmospheric losses. Although atmospheric losses depend on range, the data processing software does not take this into account. All losses are assumed to be the same for both the target and the calibration device and therefore the loss factors do not appear in the final RCS calculations. The actual target RCS is then given by

$$\sigma_{tar} = \sigma_{cal} \frac{(P_r)_{tar}}{(P_r)_{cal}} \frac{R_{tar}^4}{R_{cal}^4}. \qquad (2.4)$$

All RCS calculations in the RCSDPS software are based on this equation. The calculations of the above parameters are described in detail below.

2.2.1 RCS of Calibration Device

One of the calibration devices used by the RCSDPS software is a metallic sphere tethered to a weather balloon. The diameter of the sphere is approximately 30 cm. Since the RCS of a sphere is given by

$$\sigma_{sphere} = \pi r^2 \tag{2.5}$$

where r is the radius of the sphere, the RCS of the sphere is –11.2 dBsm.

Three other calibration devices can also be used. These include the FMF small corner reflector and the DREO medium and large corner reflectors. The RCS for these devices is not calculated, but was instead measured at the David Florida Laboratory (DFL). The values obtained from DFL for the three corner reflectors are given in Tables B1- B3 in Appendix B.

2.2.2 Calculation of Range

Since separate antennae are used for transmitting and receiving, two pulses are received after transmitting a single pulse, a leakage pulse and a reflected pulse. The actual range to the target can then be determined from the time difference between the middle of both pulses. The measured range, R_m, obtained from either the data file or the calibration file, is determined by computing the time delay from when the pulse was first transmitted to the centre of the reflected pulse. Therefore, in order to determine the actual range, a range correction, R_c, and half the pulse width, τ, must be subtracted from the measured range. The range correction accounts for the delay of the pulse from the time it was transmitted to the time it actually leaves the antenna. Since the pulse width is given in nanoseconds, the actual range is given by

$$R = R_m - 0.15\tau - R_c. \tag{2.6}$$

This equation is used for calculating the range to the calibration device as well as the target. Figure 2.1 illustrates the range calculation more clearly.

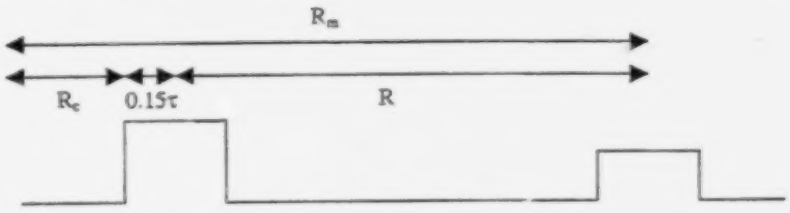


Figure 2.1: Calculation of range to target.

2.2.3 Calculation of Received Power

If the RCS is to be calculated on a pulse to pulse basis, the received power in equation 2.4 is determined by taking the amplitude of each of the I and Q samples. The power is then computed by taking the square of this amplitude. Since the received signal is multiplied by the AGC value written to the data file, it must be removed to determine the actual received power. The AGC in the data file is a power gain in dB. To remove the AGC from the received power, the AGC is converted to a straight power ratio, G, using the equation

$$G = 10^{g/10} \tag{2.7}$$

where g is the AGC value, in dB, applied to the given burst. The received power for pulse i, $P_r(i)$, in this case, is then given by

$$P_r(i) = \frac{|I_i + jQ_i|^2}{G} \tag{2.8}$$

where the values for I_i and Q_i are obtained by subtracting the I and Q offsets from the raw I and Q data.

When an RCS value is required for a single burst, the received power is calculated by computing the coherent average for the burst of data. The coherent average is computed by first determining the number of coherent pulses, n_c. This is determined by multiplying the pulse repetition frequency (PRF) with either the target coherency time, if a data file is used, or the calibration coherency time, if a calibration file is used. The coherent average is given by the equation

$$c_{ave} = \begin{cases} \frac{1}{q+1}\left[\sum_{k=0}^{q-1}\left|\frac{1}{n_c}\sum_{i=1}^{n_c} Z_{kn_c+i}\right| + \left|\frac{1}{r}\sum_{i=1}^{r} Z_{qn_c+i}\right|\right], & r>0 \\ \frac{1}{q}\sum_{k=0}^{q-1}\left|\frac{1}{n_c}\sum_{i=1}^{n_c} Z_{kn_c+i}\right|, & r=0 \end{cases} \tag{2.9}$$

where Z_i represents the complex sample of the ith pulse

$$Z_i = I_i + jQ_i \tag{2.10}$$

and q and r are determined such that the total number of pulses in the given burst, N, is given by

$$N = qn_c + r. \tag{2.11}$$

The coherency time, used to determine the number of coherent pulses, is an adjustable parameter. It represents the length of time consecutive pulses remain coherent. Consecutive pulses are coherent if the phase does not change by more than $\pi/8$ radians.

Since the coherent average for a burst is equivalent to an amplitude calculation, the power is determined by squaring the coherent average. As in the previous case, the AGC value must then be removed to determine the actual received power. The actual received power for a single burst is given by

$$P_r = \frac{(c_{ave})^2}{G}. \tag{2.12}$$

2.3 Calculating the Range Resolution Using HRR Data

High range resolution data, collected using the RCSDAS system, is used to determine the location of the major scatterers of a given target by plotting the received power versus range graph. This is accomplished using the following algorithm.

Since HRR data consists of n pulses acquired over a range of evenly spaced frequencies, $f_1, f_2, \ldots, f_n$, the data can be considered as complex frequency domain data. Taking an inverse Fourier transform of the data gives the discrete time domain data with a time interval given by

$$\Delta t = \frac{1}{B_w} \tag{2.13}$$

where B_w is the bandwidth. In this case, the bandwidth is defined as

$$B_w = f_n - f_1 \tag{2.14}$$

where the frequencies, f_1 and f_n, represent the minimum and maximum frequencies, respectively, in the frequency sweep range.

The time domain can then be converted to a range domain with range spacing, Δr, by using the equation

$$\Delta r = \frac{c\Delta t}{2} \tag{2.15}$$

where c is the speed of light. Substituting equation 2.13 gives

$$\Delta r = \frac{c}{2B_w}. \tag{2.16}$$

This range spacing defines the range axis in the power versus range graph where the range given by equation 2.6 is the centre of the range axis. The power is computed by squaring the magnitude of the result of the inverse Fourier transform.

3 RCSDPS Software

The Radar Cross Section Data Processing System (RCSDPS) is designed to analyse raw radar data obtained from the RCS Data Acquisition System (RCSDAS). Several features are included in this data processing software. These include

1. Viewing raw data of a single burst,
2. Processing a single burst,
3. Processing RCS data for an entire data set (several bursts),
4. Comparing processed RCS data and,
5. Processing high range resolution (HRR) data.

A flow chart of the RCSDPS software is given in Figure 3.1. The flow chart shows all of the options available and the programs and/or functions which are executed when each of the options are selected.

The details of how to use the software is described in [2], while the software structure for each of the above options is described in detail in the following sections.

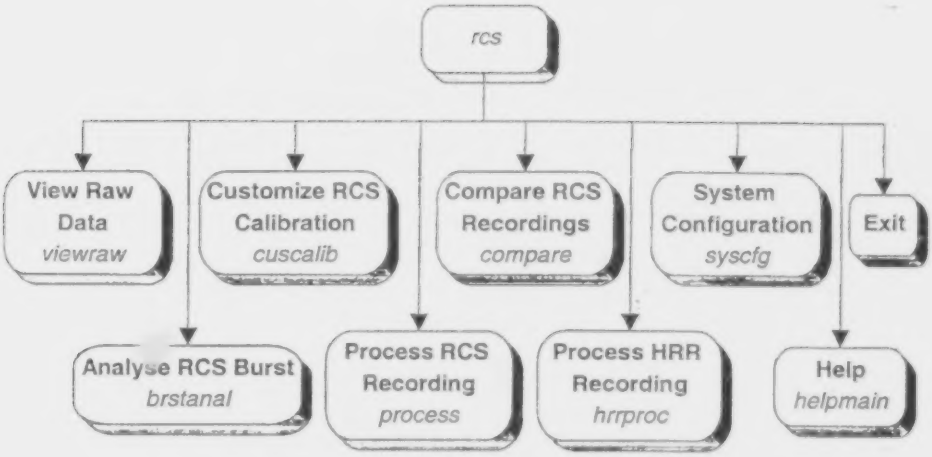


Figure 3.1: Flow chart for RCSDPS main program.

3.1 View Raw Data

The raw data for a selected burst, from either a data file or a calibration file, can be displayed by selecting the **View Raw Data** option from the RCS main menu. When this option is selected the program *viewraw* is executed. Figure 3.2 shows a flow chart for the program.

From the flow chart, it can be seen that the program *fileraw* is executed each time *viewraw* is executed. It is also executed every time a new data file is selected by selecting the **Select Raw Data File** option. The flow chart for the *fileraw* program is given in Figure 3.3. This program reads the selected file using the function *dataread*() and then calls the program *getburst*.

The program *getburst* reads the I and Q data for the specified burst using the function *getiq()* and then plots the raw data using the function *plotraw()*. Once the I and Q offsets are removed from the raw I and Q data, four graphs are displayed. The I and Q data, the phase and the amplitude are each plotted versus pulse number on separate graphs. In addition to the graphs, the coherent and non-coherent averages are computed and displayed along with other file header information. The coherent average is calculated using equation 2.9, while the non-coherent average is given by the following equation

$$n_{ave} = \frac{1}{N}\sum_{i=1}^{N}|Z_i| \tag{3.1}$$

where Z_i is the complex sample of the ith pulse and N is the total number of pulses in the burst. The flow chart for *getburst* is given in Figure 3.4.

The detailed algorithms for the functions *getiq()*, *dataread()*, and *plotraw()* are given in Appendix C.

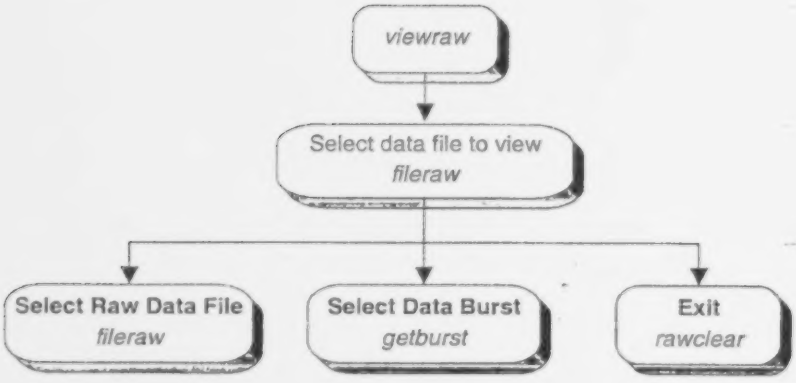


Figure 3.2: Flow chart for program *viewraw*.

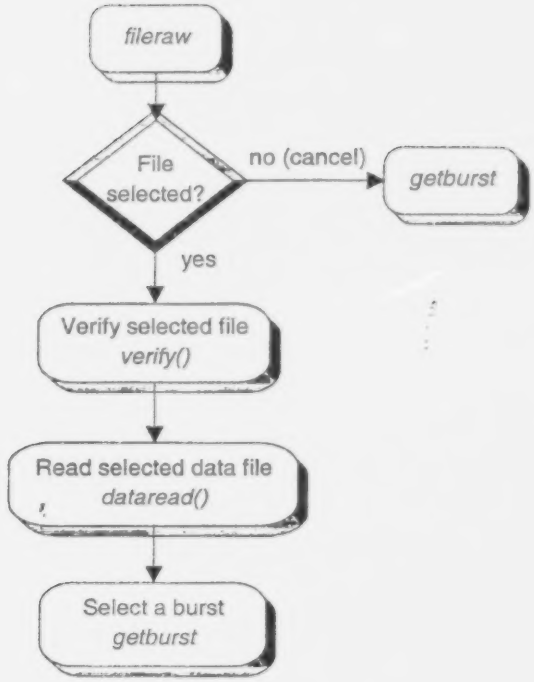


Figure 3.3: Flow chart for program *fileraw*.

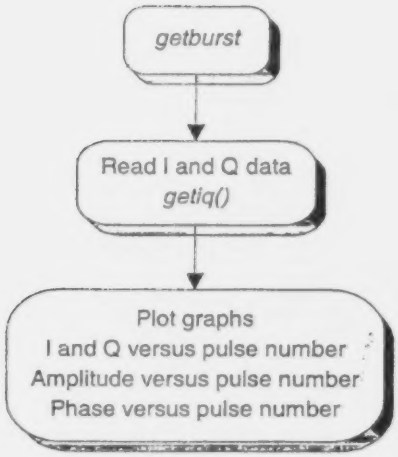


Figure 3.4: Flow chart for program *getburst*.

3.2 Analyse RCS Burst

The **Analyse RCS Burst** option computes the RCS for each pulse in the selected burst. It is executed using the program *brstanal*. Several options are included for displaying the results of the RCS calculations. These include plotting the computed RCS versus time, plotting the average RCS versus time using either a binned or sliding window algorithm, or plotting the calibrated magnitude of the Fourier transform of the I and Q data. The structure of the **Analyse RCS Burst** option is illustrated in the flow chart given in Figure 3.5.

As shown in the flow chart, the program *fileanal* is executed every time the program *brstanl* is executed. It is also executed when a new data file is selected using the **Select Data File** option. The structure for the program *fileanal* is given in Figures 3.6 and 3.7. This program reads the selected calibration file(s). If two calibration files are selected then the files are combined using the function *proccal()*. The algorithm used in *proccal()* is described in detail in Appendix C.

Selecting the **Select Data'Burst** option executes the *getbrst2* program. The flow chart for *getbrst2* is given in Figure 3.8. This program executes the *getiq()* function to read the I and Q data for the selected burst and then executes the *burstrcs()* function to compute the RCS for each pulse in the burst. The *burstrcs()* function also displays the RCS versus time graph and computes the coherent and non-coherent averages, using equations 2.9 and 3.1 respectively. An RCS value for the entire burst is also calculated using equations 2.4 and 2.12.

The average RCS versus time graph is displayed using the program *avgmenu*. This program creates a submenu with two options for computing the average RCS. The two algorithms are binned averaging and sliding window averaging. Both algorithms are executed using the function *avgbrcs()*. This is illustrated in the flow chart in Figure 3.5. The binned averaging algorithm divides the number of pulses in the given burst into contiguous segments of a user specified length with the remaining pulses forming a smaller segment ([2] p. 17). Each of the segments is then coherently averaged and an average RCS value is determined using equations 2.12 and 2.4. The sliding window averaging algorithm coherently averages the specified number of pulses using a sliding window where the window slides by one pulse at a time. The average RCS is then calculated using the same equations as for the binned averaging algorithm.

The **Calibrated FFT** option executes the function *fftplt()*. This function computes the Fourier transform of the complex I and Q samples for a given burst. Taking the magnitude and squaring the result of the Fourier transform gives the power spectral density, or the measure of the power at different frequencies. Using these values for the received power of the target, $(P_r)_{tar}$ in equation 2.4, illustrates how the RCS is distributed across the different frequency components.

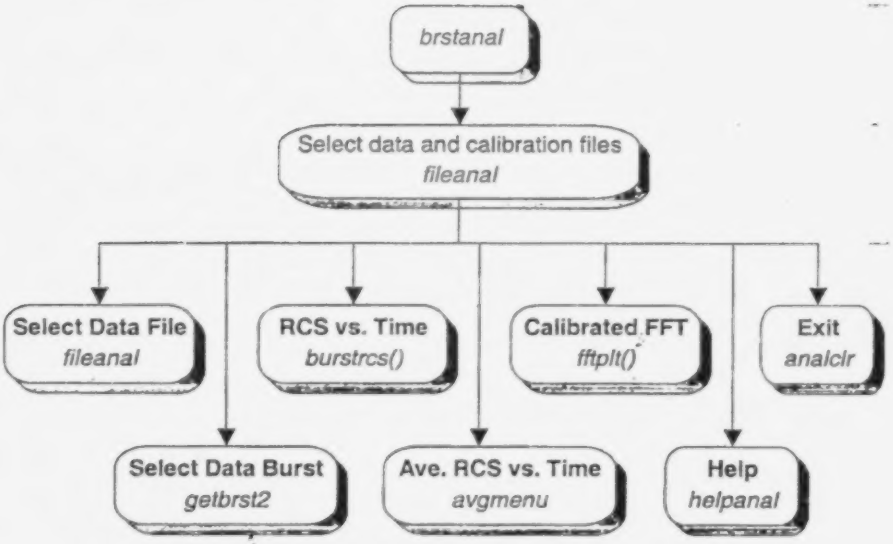


Figure 3.5: Flow chart for program *brstanal*.

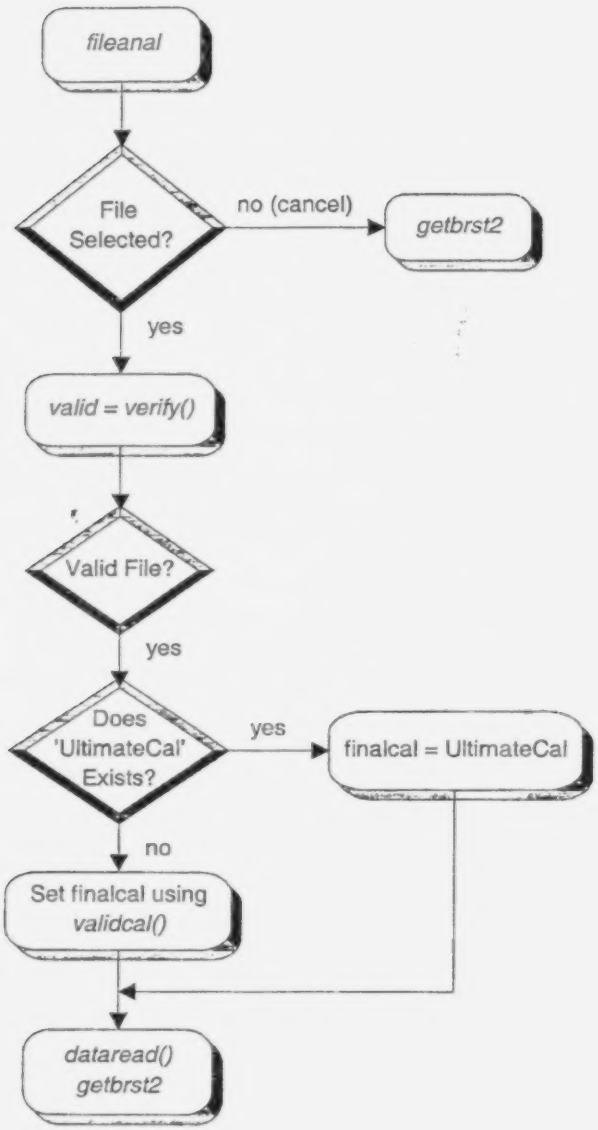


Figure 3.6: Flow chart for program *fileanal*.

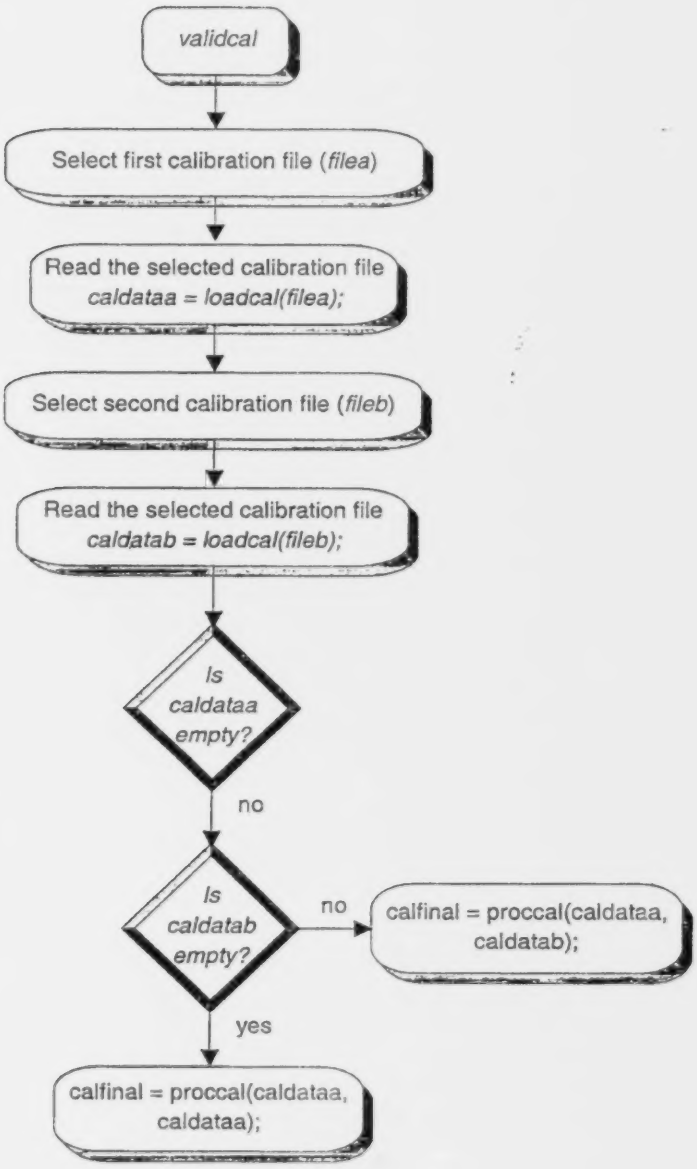


Figure 3.7: Flow chart for program *validcal*.

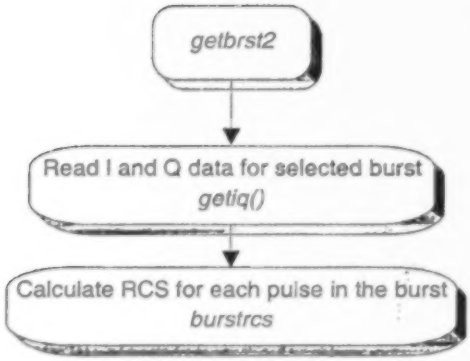


Figure 3.8: Flow chart for program *getbrst2*.

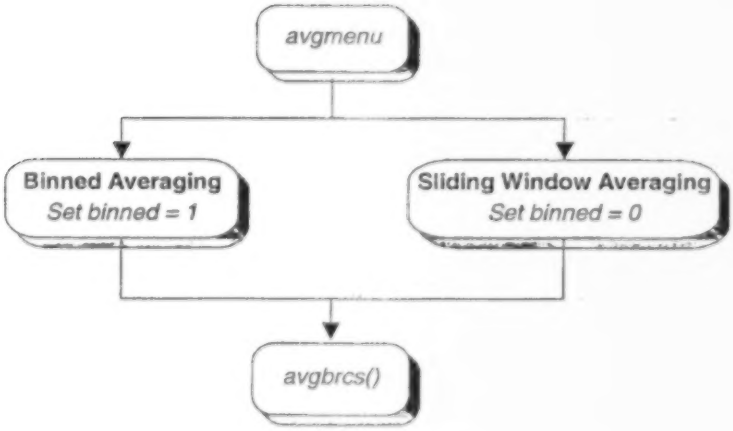


Figure 3.9: Flow chart for program *avgmenu*.

3.3 Customize RCS Calibration

This option is provided so that the calibration values, such as the AGC and the coherent average can be adjusted in order to improve the accuracy of the RCS calculations. The new values are stored in a manual calibration file, which has the same name as the calibration file except the prefix is changed from 'C' to 'M'. This option executes the *cuscalib* program. The flow charts for the program are given in Figures 3.10 and 3.11.

The calibration process begins by using a data file where the target is the calibration device. Using this data file, the RCS is calculated for the target using equation 2.4. This value is then compared to the known RCS value. If the value is incorrect, either the AGC or the coherent average is adjusted and the RCS is calculated again. This process continues until the calculated RCS value corresponds to the actual RCS value.

Although this iterative method can be used to optimize the calibration parameters, the optimal values can be calculated directly. This is accomplished by selecting a burst where the power and phase are fairly constant. Then, the AGC value should be set to the actual AGC and the coherent average should be set to the square root of the average power of the pulses in the given burst. This will result in the average RCS for the burst to be equal to the actual RCS.

The new AGC and coherent average values are set using the **Set Calib. Values** option. Once the values are set, for each of the parameter combinations available, they are saved in the manual calibration file by selecting the **Save Calib. Values** option.

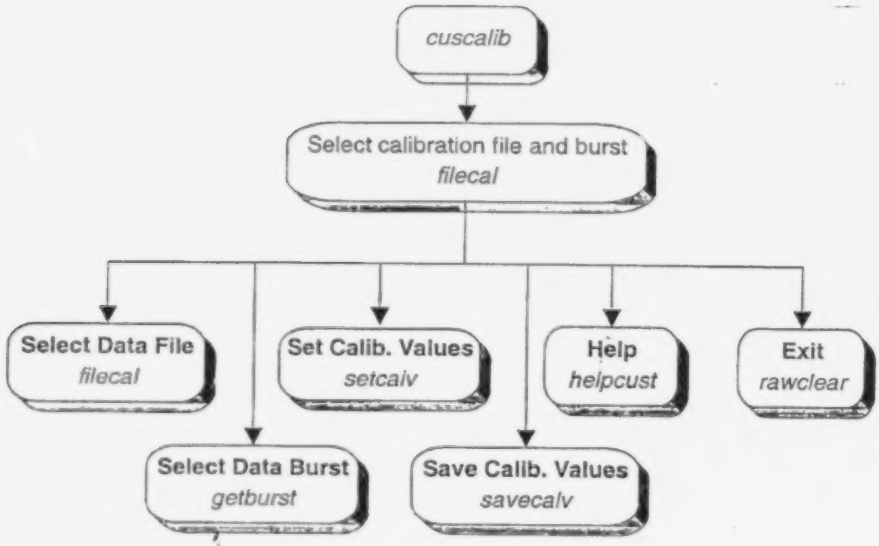


Figure 3.10: Flow chart for program *cuscalib*.

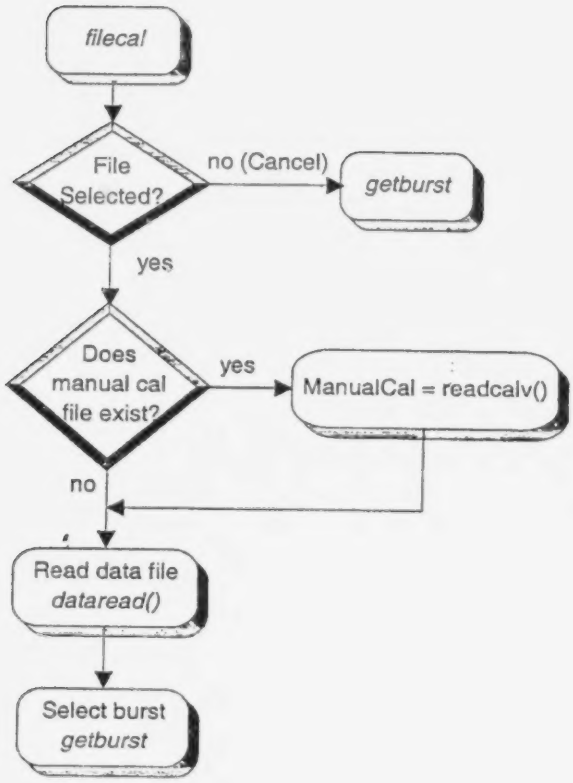


Figure 3.11: Flow chart for program *filecal*.

3.4 Process RCS Recording

The **Process RCS Recording** option determines an RCS value for each burst in the selected data file. It executes the program *process*. Figure 3.12 shows the flow chart for the program. As shown in the flow chart, the program *process* executes the program *fileproc*, which reads the selected data file and calibration file(s) and computes the RCS for each burst using equations 2.12 and 2.4. After selecting all bursts with the same default parameters, the RCS is computed for each pulse in each of the selected bursts using the program *makercs*. This program uses the equations 2.8 and 2.4. These computed RCS values are not plotted, but are stored and can be plotted in the **Summary Pages** option. The single RCS value computed for each burst, however, is plotted versus time using the program *proctime*. This is illustrated in the flow chart for *fileproc* given in Figure 3.13. The flow chart for *makercs* is given in Figure 3.14. This program calls the functions *getiq()* and *pulsercs()* for each of the selected bursts. The function *getiq()* reads the I and Q data for a given burst, while *pulsercs()* calculates the RCS for each pulse in the burst.

The **Select Parameters** option executes the program *procparm* after setting the variable *choice* to 1. This option determines all the bursts with frequency, pulse width, transmit polarization and receive polarization equal to the specified values. The program *procparm* calculates the RCS for each pulse in each of the selected bursts using the program *makercs*. It also processes the telemetry data, if it exists, using the function *bindata()*. The RCS for each burst is then plotted versus time using either the program *proctime* or the program *comptime*. Since *procparm* is also called from the **Compare RCS Recording** option, which is described in a later section, it must determine if the process option or the compare option is selected. If the compare option is selected *comptime* is executed otherwise *proctime* is executed. This is illustrated in the flow chart for *procparm* given in Figure 3.15. The program *proctime* is also executed when the **RCS vs. Time** option is selected.

The **RCS vs. Angle** option is either enabled when telemetry data exists for the data file selected or when the selected data file contains more than four event marks. An event occurs when the bearing of the ship is recorded. The bearing values following each event mark are recorded in degrees and are processed using the function *bindata()*. This function assigns an RCS value to each degree between the event marks. If the number of degrees between event marks is greater than the number of data points, the RCS values are interpolated to obtain an RCS value for each degree. Otherwise, if the number of degrees is less than the number of data points, the RCS values are averaged. The function *bindata()* is also used to process telemetry data. Telemetry data contains the bearing, in degrees, of the ship and the time at which the bearing was recorded. When the function *bindata()* is used with telemetry data, an RCS value is assigned to each bearing value by averaging the RCS values between the times the current bearing and the next bearing are recorded.

When the **RCS vs. Angle** option is selected, the program *procrcs* is executed. This option is used to show how the RCS changes as the bearing of the ship target changes. Two options are available for plotting the RCS. A raw binned RCS plot and a sliding window average plot. The sliding window average is obtained by averaging the raw binned RCS plot using a specified window size. Both algorithms are described on page 26 of [2]. The flow chart describing the program is given in Figure 3.16.

The **Histogram** option is used to plot the histogram of the RCS values for each pulse in each of the selected bursts. MATLAB's histogram function is used with the number of bins fixed to twenty. The bursts are selected using the **Select Parameters** option. Initially, a cumulative distribution curve is plotted on the same axis as the histogram. Once the histogram option is selected, the **Split** option is enabled. This plots the histogram and the cumulative distribution curve on separate graphs by executing the program *procsplt*. The **Combine** option is used to plot both graphs on the same axis. It uses the program *proccomb*.

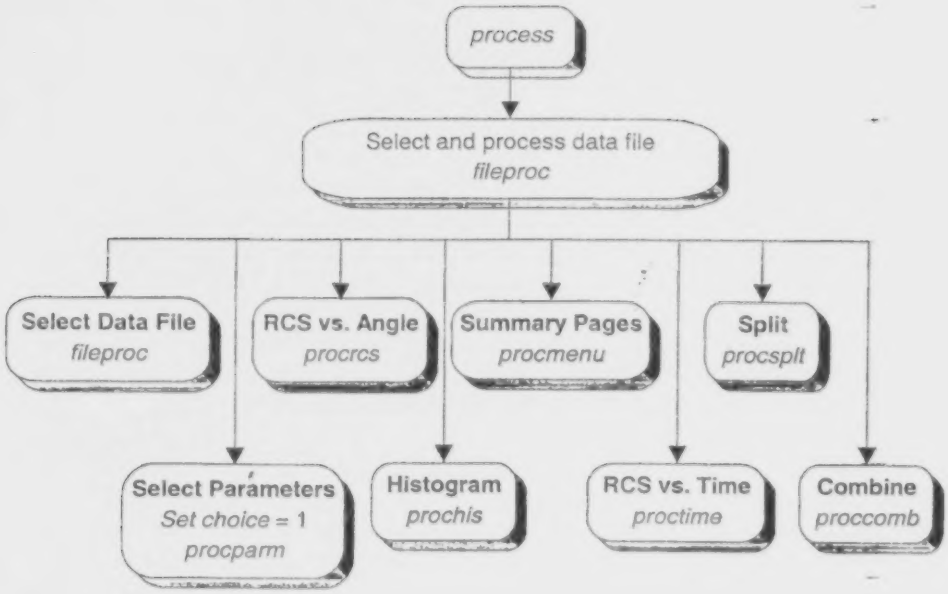


Figure 3.12: Flow chart for program *process*.

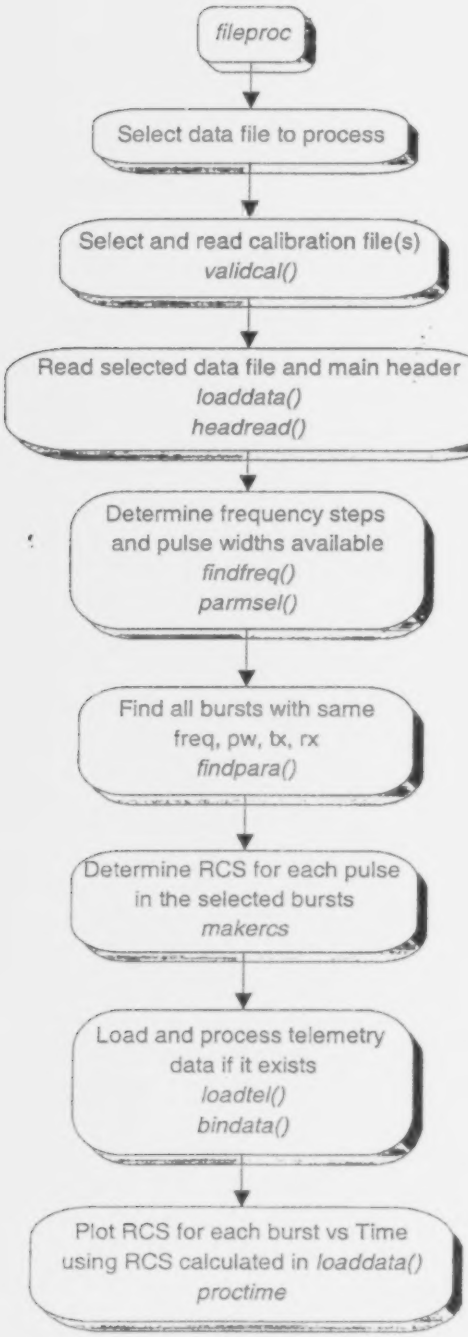


Figure 3.13: Flow chart for program *fileproc*.

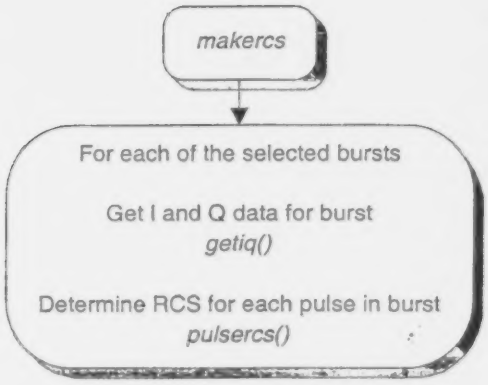


Figure 3.14: Flow chart for program *makercs*.

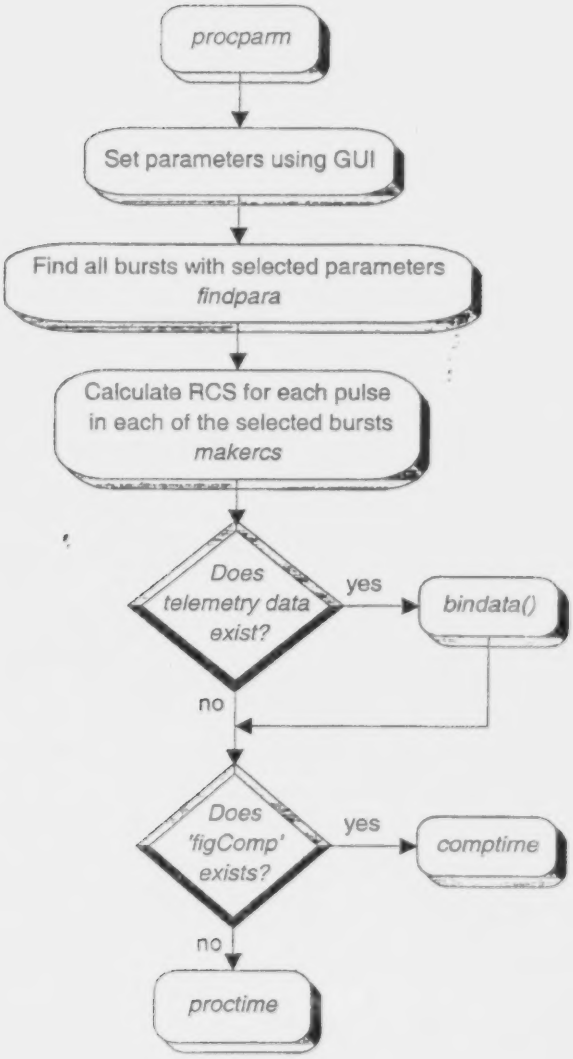


Figure 3.15: Flow chart for program *procparm*.

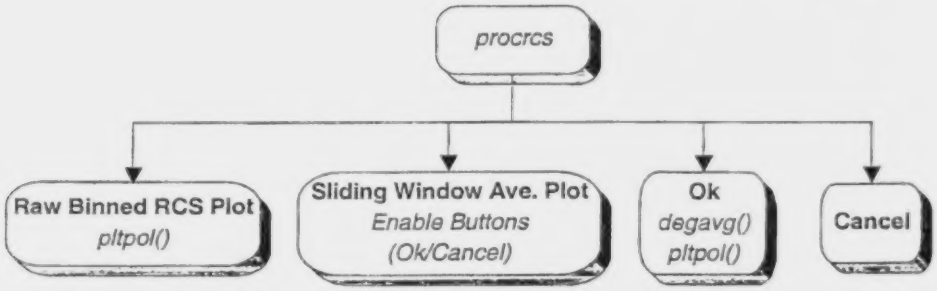


Figure 3.16: Flow chart for program *procrcs*.

The **Summary Pages** option executes the program *procmenu*. Three options are available for producing a summary page, **Multiple Parameter Plots**, **Complete Summary Page** and **Report Summary Page**. The **Multiple Parameter Plots** and **Complete Summary Page** options execute the programs *multparm* and *procsum1*, respectively. The **Report Summary Page** option executes either *prcsmenu*, if telemetry data exists, or *procsum2* if there is no telemetry data. This is illustrated in the flow chart for *procmenu* given in Figure 3.17.

The program *multparm* is used for setting the parameters used for plotting the RCS for each burst versus time. Once the parameters are set, it calls the program *multplot* to plot either a single curve for a specified frequency or multiple curves if the all frequencies option is selected from the **Select Parameters** option. The flow chart for *multparm* is given in Figure 3.18.

The program for the **Complete Summary Page** option, *procsum1*, displays the histogram of the RCS values, calculated for the bursts selected using the **Select Parameters** option. A cumulative distribution curve is also plotted on the same axis. In addition to the graphs, the average RCS is calculated and displayed along with the system configuration parameters and file header information. If telemetry data exists, then a polar plot of the RCS versus the angle of the ship is also displayed. The flow chart for *procsum1* is given in Figure 3.19.

The Report Summary Page, created using the program *procsum2*, displays the RCS versus time graph as well as the histogram of the RCS and the cumulative distribution curve. In this case, the variable *togglerun* is set to 2. If telemetry data exists, the program *prcsmenu* sets the variable *togglerun* to either 0 or 1, depending on which algorithm is used for processing the data. Then, it executes the program *procsum2*. This is illustrated in the flow charts for *procsum2* and *prcsmenu* in Figures 3.20 and 3.21, respectively.

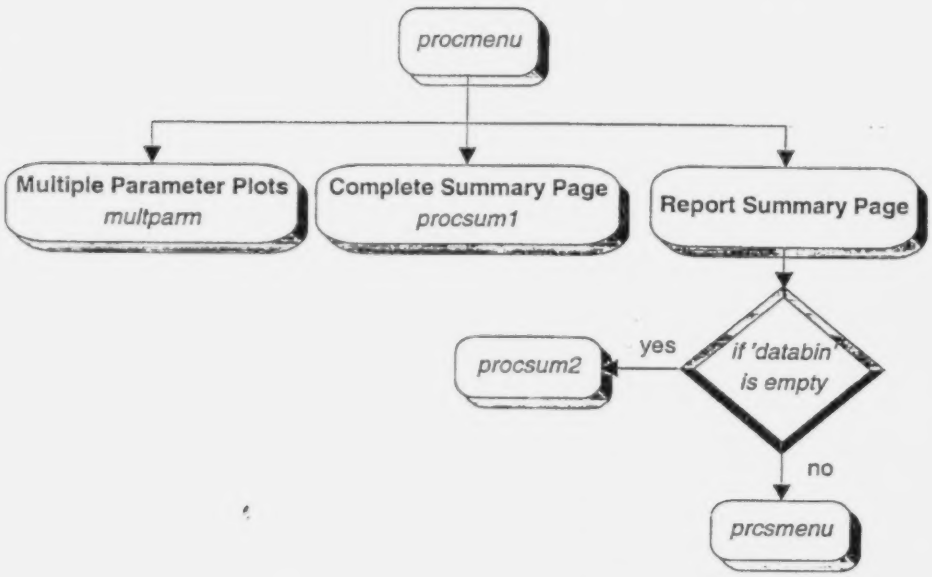


Figure 3.17: Flow chart for program *procmenu*.

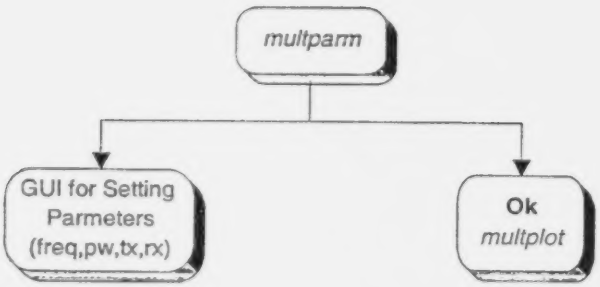


Figure 3.18: Flow chart for program *multparm*.

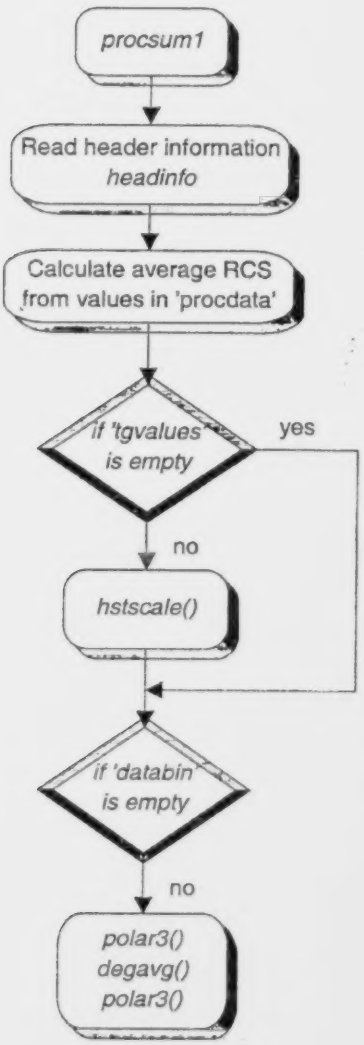


Figure 3.19: Flow chart for program *procsum1*.

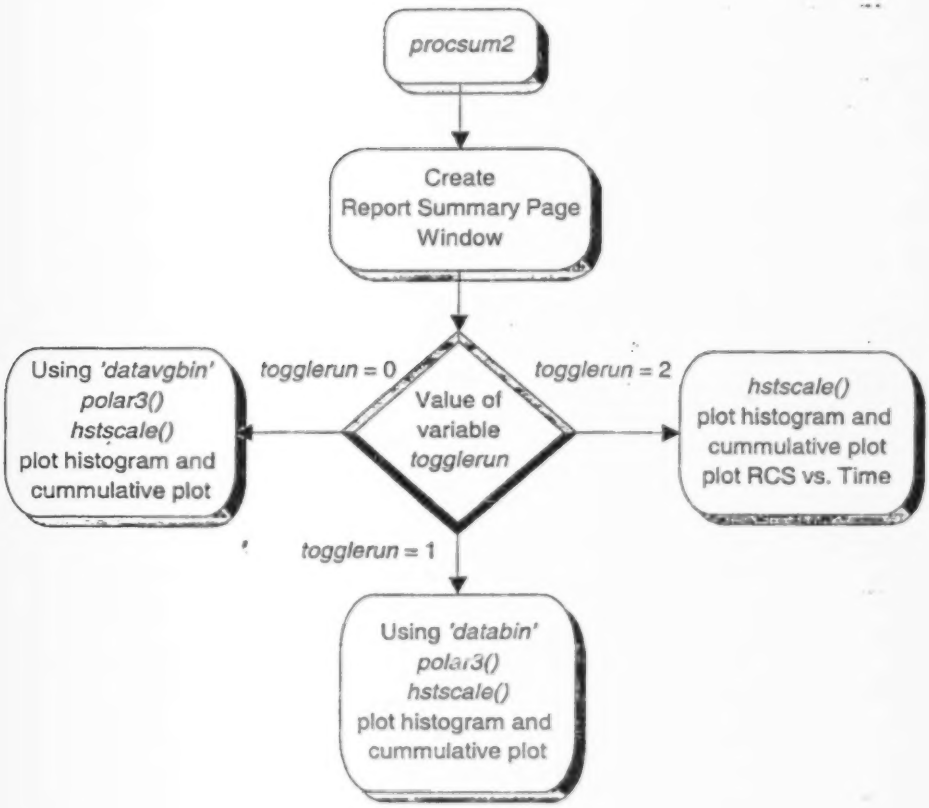


Figure 3.20: Flow chart for program *procsum2*.

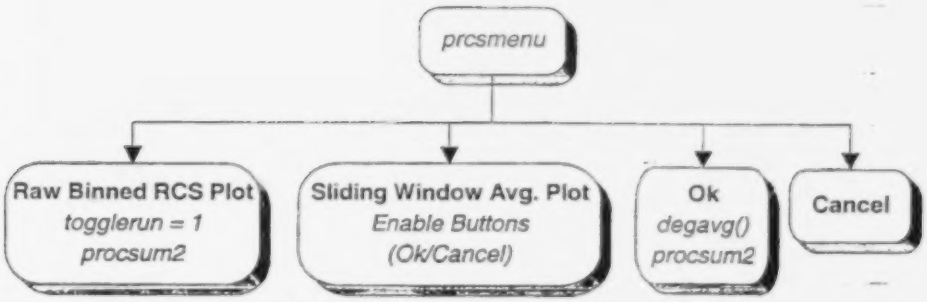


Figure 3.21: Flow chart for program *prcsmenu*.

3.5 Compare RCS Recording

The **Compare RCS Recording** option, from the RCS main menu, is used to compare the RCS calculations for two different data files. It is executed using the program *compare*. All of the options in the **Process RCS Recording** option such as the various RCS plots, the summary pages and the histogram are included in this option. Figure 3.22 shows the flow chart for the program *compare*.

The programs used in this option are almost identical to those used in the **Process RCS Recording** option. The only difference is that the operations and calculations are applied to two different data files for comparison. Therefore, the related programs are not described in detail in this section. However, the structure of the code is illustrated by the flow charts given in Figures 3.23 to 3.29.

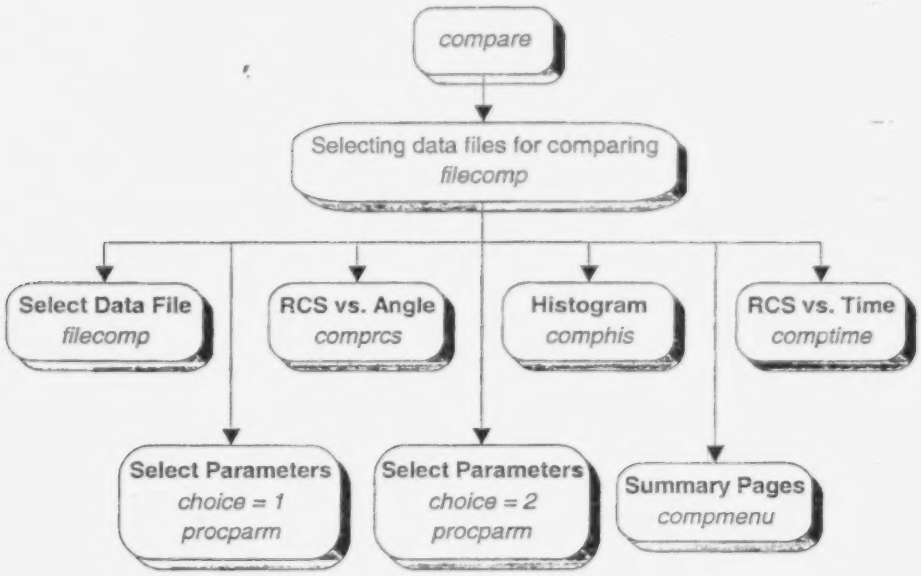


Figure 3.22: Flow chart for program *compare*.

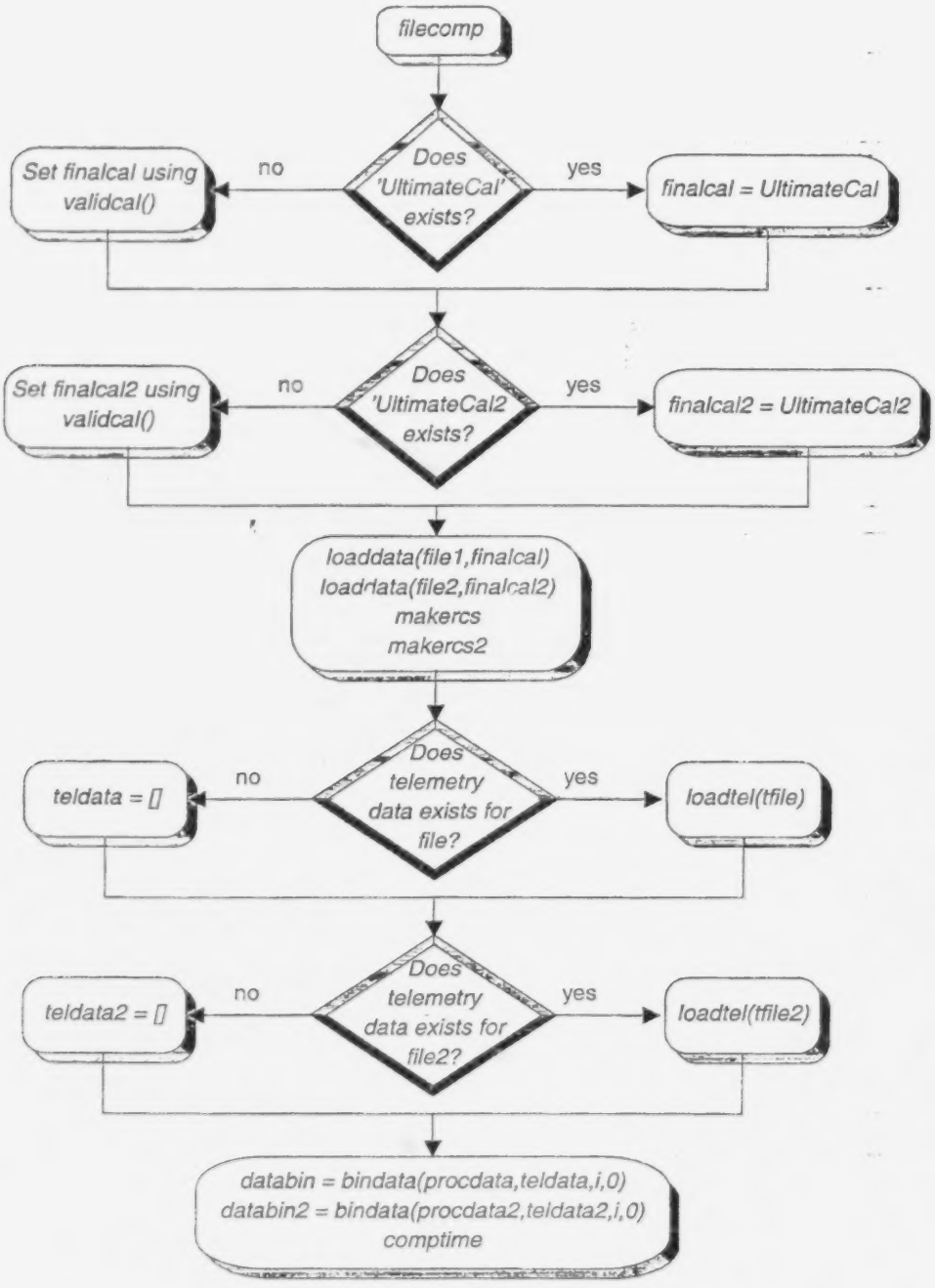


Figure 3.23: Flow chart for program *filecomp*.

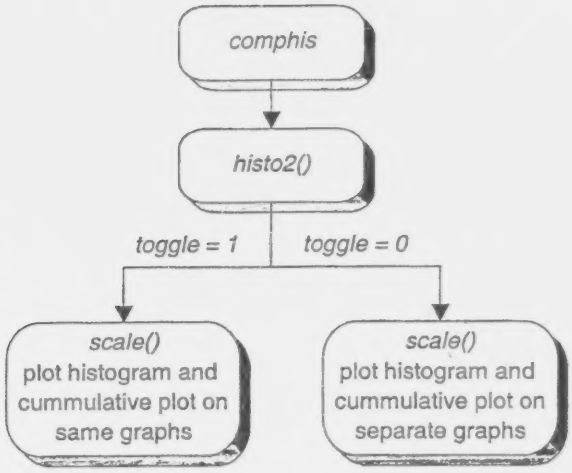


Figure 3.24: Flow chart for program *comphis*.

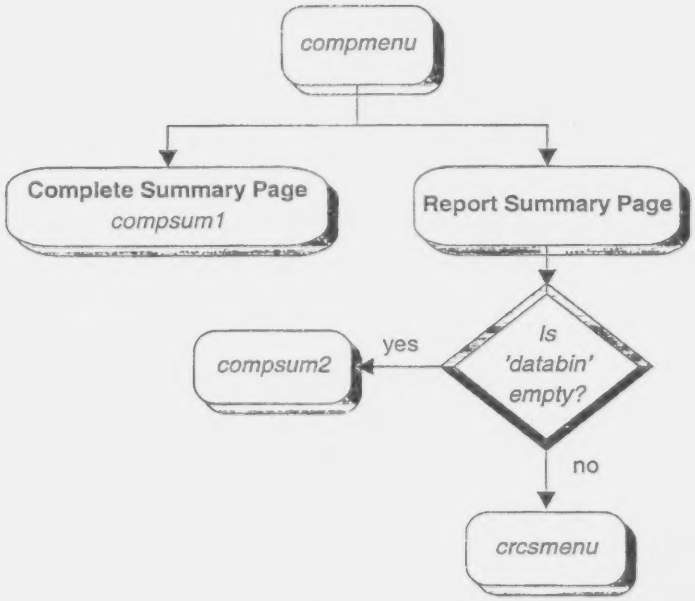


Figure 3.25: Flow chart for program *compmenu*.

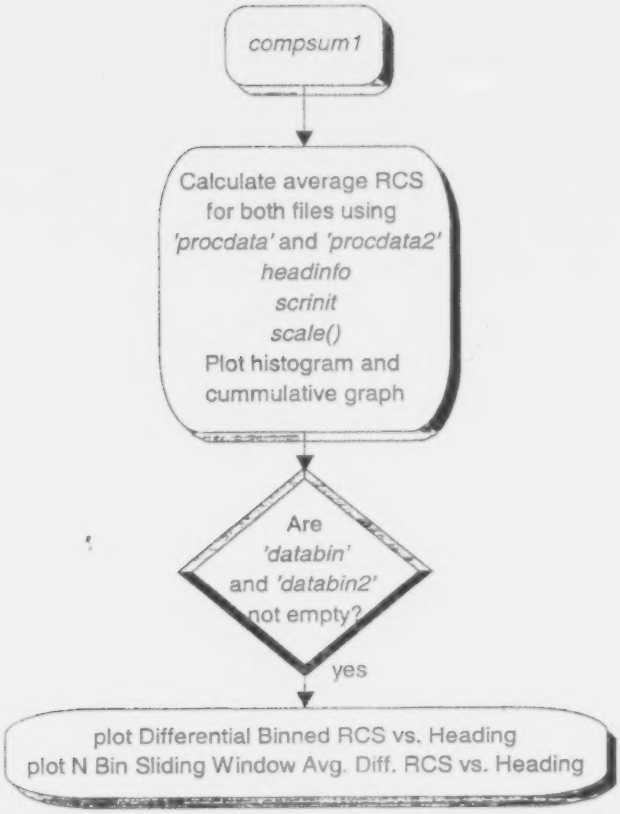


Figure 3.26: Flow chart for program *compsum1*.

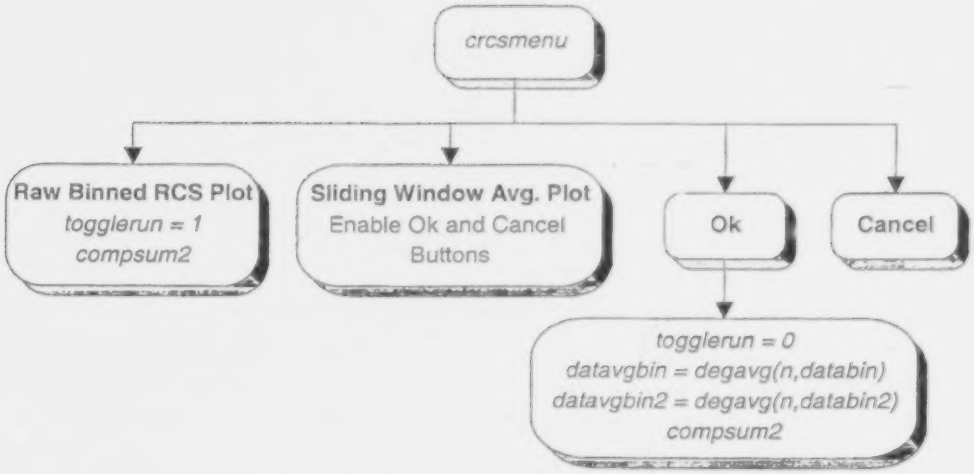


Figure 3.27: Flow chart for program *crcsmenu*.

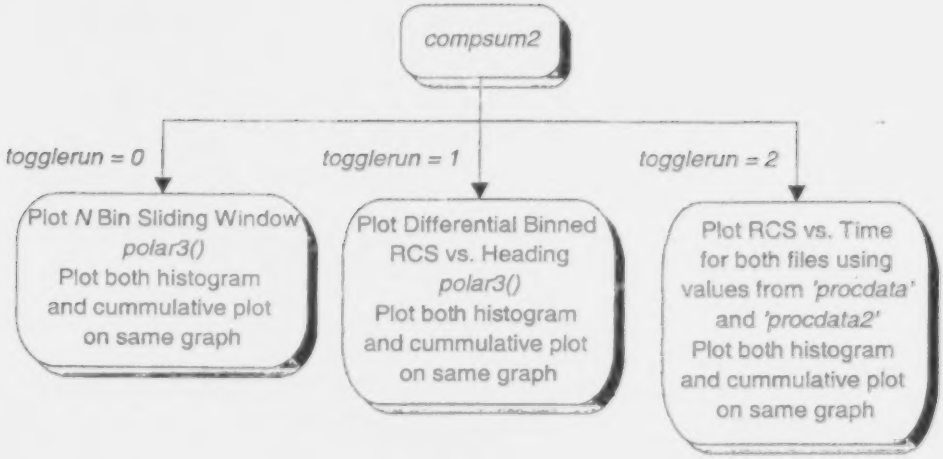


Figure 3.28: Flow chart for program *compsum2*.

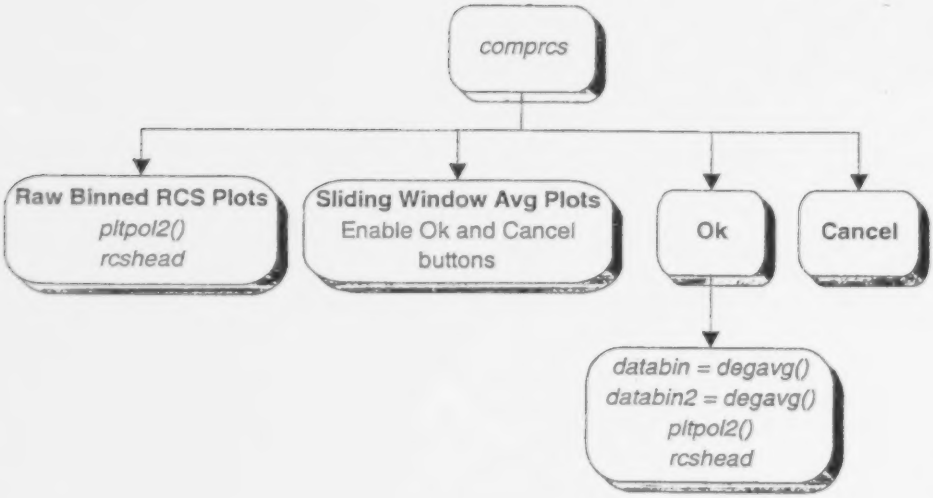


Figure 3.29: Flow chart for program *comprcs*.

3.6 Process HRR Recording

The purpose of the **Process HRR Recording** option in the RCS main menu is to process HRR data using the algorithm described in section 2.3. It is executed using the program *hrrproc*. Figure 3.30 gives a flow chart illustrating the structure of the program. As shown in the flow chart, a number of options are available. However, the main purpose is to plot the power versus range graph to obtain an accurate range profile to the given target.

As indicated by the flow chart, the program *hrrproc* executes the program *filehrr* to select an HRR data file. Figure 3.31 gives a flow chart for *filehrr*. Once a file is selected, it is read using the function *readhrr()*. The format for HRR data is the same as an RCS data recording or a calibration file used in the RCS data processing.

The variables, *startHRR* and *stopHRR*, are used to define the first and last pulse to be used in the HRR data processing. They also define the bandwidth given by equation 2.14.

After reading the selected file, the program *filehrr* executes the program *gethrr* to select a burst from the given data file. The flow chart for *gethrr* is given in Figure 3.32. This program executes the function *getiq()* to read the I and Q data, then executes the function *plothrr()* to plot four graphs. These include the power versus range graph, as well as, the I data values versus frequency, the phase versus frequency and the power versus frequency.

The **Select IFFT Limits** option is executed using the program *setifft*. It is used to select a range of pulses for HRR processing by setting the variables *startHRR* and *stopHRR*. These limits should only be changed if the phase versus frequency plot is non-linear. If this is the case, the data containing the linear portion should be selected by setting the variables *startHRR* and *stopHRR* appropriately. Once the new limits are set, the program *setifft* executes the function *plothrr()* to process the HRR data again and to re-display the new graphs. The flow chart for *setifft* is given in Figure 3.33.

The **Full Screen HRR** option is used to plot the power versus range graph without the other three graphs. This is accomplished by setting the global variable *HRRfull* to one and executing the function *plothrr()*. If *HRRfull* is set to zero, *plothrr()* will plot all four graphs.

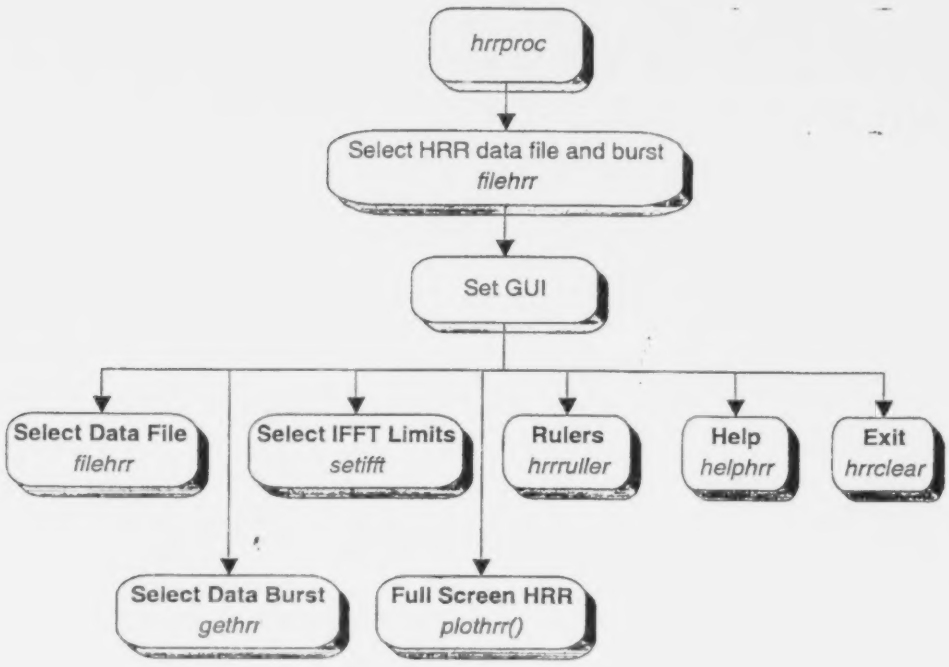


Figure 3.30: Flow chart for program *hrrproc*.

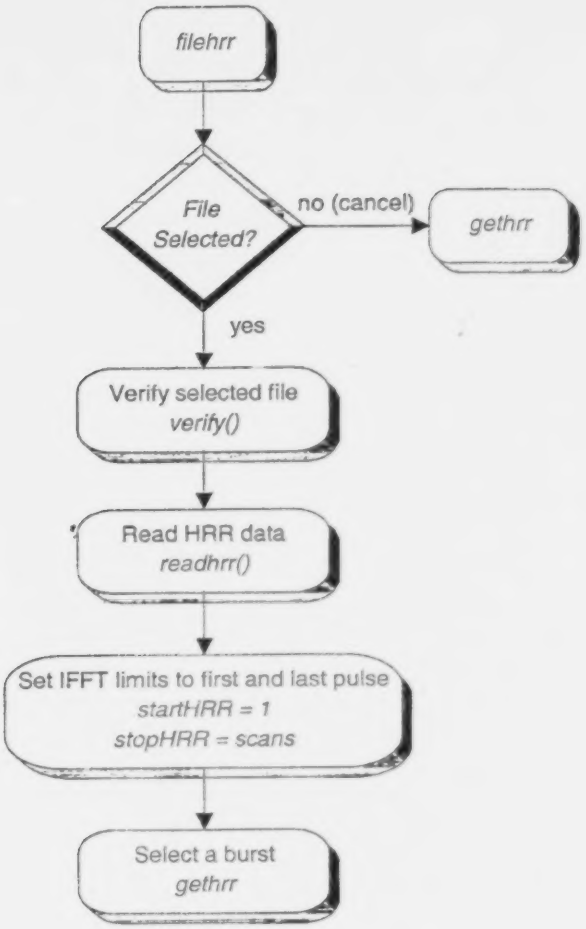


Figure 3.31: Flow chart for program *filehrr*.

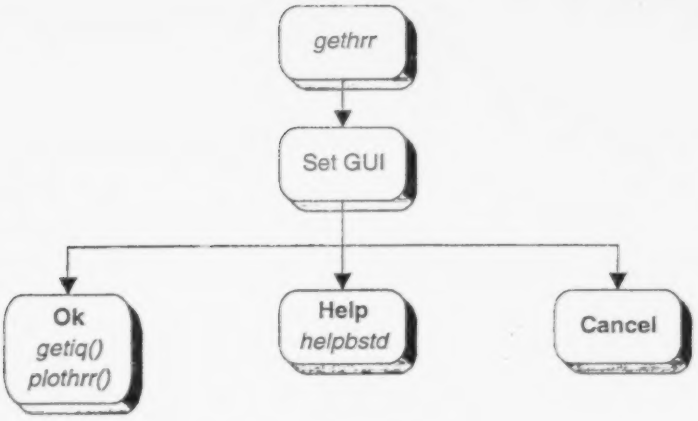


Figure 3.32: Flow chart for program *gethrr*.

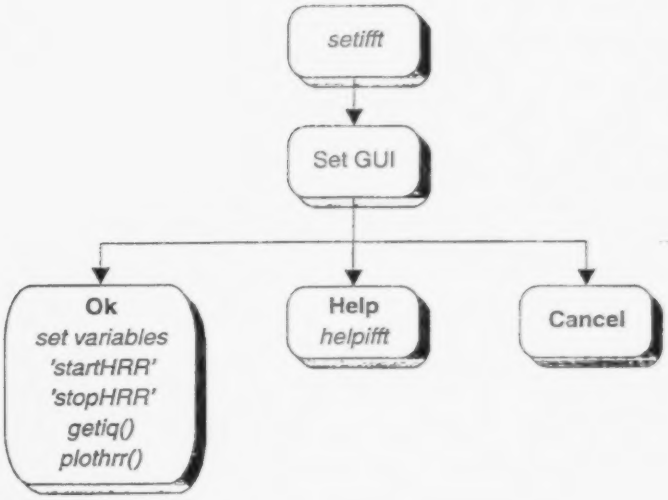


Figure 3.33: Flow chart for program *setifft*.

3.7 System Configuration

The **System Configuration** option is used to select the calibration target and to set the system configuration parameters such as the coherency time for both, the calibration device as well as the target, the coherency tolerance, the range correction value and the I and Q offsets. This option is executed using the program *syscfg*. The flow chart for the program is given in Figure 3.34.

The program *savecfg* saves the new system configuration parameters into the file *system.cfg*, while the program *copycfg* copies the file corresponding to the calibration target to the file *calibrcs.m*.

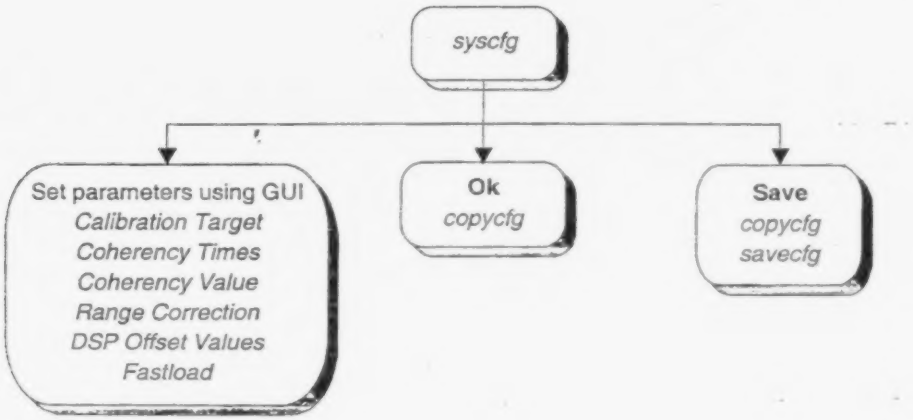


Figure 3.34: Flow chart for program *syscfg*.

4 References

[1] Radar Cross Section High Range Resolution Data Acquisition System. Technical Reference Manual. Version 3.00. Atlantis Scientific Inc. March, 1996.

[2] Radar Cross Section Data Processing System. Software User's Guide and Reference Manual. Version 3.1. Atlantis Scientific Inc. March, 1998.

5 Appendix A: Sample Calibration File

The following figure gives an example of part of a calibration file. It shows the file header information, the burst header for the first burst followed by the I and Q data for the first burst.

```
14
Recording Number:  0025     Date:  980728     Time:  095637
Description:  _Calibration Data                              Target Type:config0
Recorded by:          Recorded scans:  00500

System configuration:

PRF: 10.00 KHz
Frequencies (GHz): 9.00 15.00 Samples Per Burst: 100
Trig Rising Edge     Trig thresh:  050
Targ thresh:   000    2V Range     DC offset:       000
Manual Gate: 1  Gate start:    00247 m     Gate end:     00247 m     Gate width:      00000 m
--------------------------------------------------------------------------------

T901634197.799 Freq15.0000 Pwr13.00 PW 600 AGC35.13 Attn 0.00 Range  247 TxH RxH Scans 100 Noise0
28203 29593
27959 30256
28076 30211
27982 30127
28022 30547
27826 30378
28271 30454
27957 31108
27855 31262
28039 31144
27967 31648
27832 31925
27843 31614
27991 31636
27973 32330
28191 32415
28208 32156
28163 33041
28367 33065
28661 33218
28614 33347
28397 33431
28720 33362
```

6 Appendix B: RCS for DREO Corner Reflectors

Frequency (GHz)	RCS (HH) (dBsm)	RCS (VV) (dBsm)	Frequency (GHz)	RCS (HH) (dBsm)	RCS (VV) (dBsm)
8.0	-0.68	0.93	12.1	3.07	3.25
8.1	-1.19	0.80	12.2	2.72	2.91
8.2	-1.66	0.06	12.3	2.37	2.72
8.3	-2.15	-0.50	12.4	1.88	2.57
8.4	-2.60	-0.85	12.5	1.70	2.26
8.5	-2.92	-1.19	12.6	1.32	1.90
8.6	-3.05	-1.57	12.7	0.80	1.68
8.7	-3.17	-2.23	12.8	0.80	1.82
8.8	-3.16	-2.70	12.9	1.22	2.27
8.9	-2.74	-2.72	13.0	1.74	2.74
9.0	-1.96	-1.93	13.1	2.61	3.04
9.1	-1.07	-1.30	13.2	3.25	3.27
9.2	-0.38	-0.68	13.3	3.42	3.64
9.3	0.08	-0.50	13.4	3.77	4.13
9.4	0.59	-0.08	13.5	4.05	4.53
9.5	1.22	0.46	13.6	4.21	4.83
9.6	1.67	1.14	13.7	4.51	5.00
9.7	1.77	1.68	13.8	4.32	5.01
9.8	1.66	1.89	13.9	3.95	5.06
9.9	1.44	1.64	14.0	4.04	5.17
10.0	1.07	1.39	14.1	4.22	5.14
10.1	0.65	1.17	14.2	4.09	4.92
10.2	0.29	1.05	14.3	3.48	4.60
10.3	0.21	0.82	14.4	2.86	4.38
10.4	-0.97	0.36	14.5	2.67	4.19
10.5	-1.66	-0.14	14.6	2.79	3.61
10.6	-1.90	-0.72	14.7	3.10	2.75
10.7	-1.84	-0.99	14.8	3.28	2.40
10.8	-1.59	-0.95	14.9	3.29	2.83
10.9	-1.09	-0.44	15.0	3.54	3.20
11.0	-0.54	-0.21	15.1	3.86	3.08
11.1	-0.11	0.14	15.2	4.32	3.89
11.2	0.49	0.62	15.3	4.95	4.23
11.3	1.35	1.36	15.4	5.34	4.95
11.4	2.09	1.87	15.5	5.45	5.16
11.5	2.48	2.37	15.6	5.61	5.39
11.6	2.76	2.73	15.7	5.76	5.67
11.7	3.02	3.11	15.8	5.57	6.13
11.8	3.18	3.38	15.9	5.35	6.16
11.9	3.28	3.49	16.0	5.25	5.92
12.0	3.27	3.33			

Table B1: RCS values obtained from David Florida Laboratory for small corner reflector.

Frequency (GHz)	RCS (HH) (dBsm)	RCS (VV) (dBsm)	Frequency (GHz)	RCS (HH) (dBsm)	RCS (VV) (dBsm)
8.0	5.03	5.75	12.1	8.39	8.83
8.1	5.73	6.31	12.2	8.99	9.44
8.2	5.98	6.75	12.3	9.45	9.84
8.3	6.55	7.08	12.4	9.69	9.96
8.4	7.11	7.28	12.5	10.08	9.95
8.5	7.00	7.49	12.6	10.03	9.92
8.6	6.85	7.53	12.7	9.50	9.80
8.7	6.05	7.05	12.8	9.06	9.70
8.8	5.12	6.40	12.9	8.84	9.60
8.9	4.87	5.86	13.0	8.64	9.37
9.0	4.66	5.48	13.1	8.66	8.96
9.1	4.78	5.30	13.2	8.59	8.59
9.2	5.44	5.50	13.3	8.27	8.41
9.3	5.91	6.06	13.4	8.38	8.56
9.4	6.63	6.84	13.5	8.68	8.91
9.5	7.51	7.74	13.6	9.06	9.32
9.6	7.93	8.43	13.7	9.71	9.70
9.7	8.24	8.81	13.8	9.91	10.01
9.8	8.13	8.85	13.9	9.97	10.32
9.9	7.75	8.67	14.0	10.47	10.63
10.0	7.53	8.46	14.1	10.92	10.75
10.1	7.06	8.12	14.2	10.94	10.71
10.2	6.88	7.66	14.3	10.34	10.61
10.3	6.90	7.22	14.4	9.63	10.52
10.4	6.52	6.95	14.5	9.30	10.28
10.5	6.48	6.93	14.6	9.24	9.82
10.6	6.89	7.16	14.7	9.40	9.47
10.7	7.25	7.75	14.8	9.51	9.39
10.8	7.78	8.50	14.9	9.53	9.53
10.9	8.25	9.12	15.0	9.85	9.86
11.0	8.40	9.44	15.1	10.26	10.17
11.1	8.68	9.55	15.2	10.77	10.47
11.2	8.84	9.55	15.3	11.39	10.70
11.3	9.01	9.39	15.4	11.71	11.28
11.4	9.13	9.14	15.5	11.63	11.40
11.5	8.73	8.85	15.6	11.57	11.06
11.6	8.19	8.50	15.7	11.45	10.90
11.7	7.69	8.14	15.8	10.91	10.75
11.8	7.50	7.86	15.9	10.42	10.61
11.9	7.61	7.92	16.0	10.22	10.55
12.0	7.95	8.31			

Table B2: RCS values obtained from David Florida Laboratory for medium corner reflector.

Frequency (GHz)	RCS (HH) (dBsm)	RCS (VV) (dBsm)	Frequency (GHz)	RCS (HH) (dBsm)	RCS (VV) (dBsm)
8.0	10.89	11.06	12.1	13.75	13.64
8.1	10.09	10.75	12.2	13.29	13.52
8.2	9.28	10.06	12.3	13.17	13.58
8.3	8.93	9.45	12.4	13.08	13.52
8.4	9.03	9.08	12.5	12.63	13.07
8.5	9.78	9.33	12.6	12.31	12.66
8.6	11.02	9.99	12.7	12.24	12.40
8.7	12.11	10.75	12.8	12.50	12.41
8.8	12.36	11.03	12.9	12.89	12.47
8.9	12.01	10.98	13.0	12.99	12.35
9.0	11.93	11.41	13.1	13.15	12.60
9.1	11.51	11.49	13.2	13.45	13.19
9.2	10.60	10.85	13.3	13.45	13.57
9.3	9.83	10.00	13.4	13.06	13.49
9.4	9.77	9.67	13.5	12.71	13.20
9.5	10.53	9.84	13.6	12.94	13.31
9.6	11.44	9.87	13.7	13.32	13.50
9.7	12.10	10.11	13.8	13.52	13.44
9.8	12.87	11.04	13.9	13.76	13.39
9.9	13.57	12.15	14.0	14.07	13.62
10.0	13.52	12.71	14.1	14.31	14.08
10.1	12.88	12.68	14.2	14.16	14.23
10.2	12.27	12.51	14.3	13.69	13.85
10.3	11.67	12.04	14.4	13.34	13.41
10.4	11.17	11.29	14.5	13.22	13.18
10.5	10.92	10.45	14.6	13.22	13.10
10.6	10.95	9.90	14.7	13.28	13.19
10.7	11.74	10.40	14.8	13.55	13.61
10.8	12.84	11.54	14.9	14.06	14.22
10.9	13.44	12.49	15.0	14.47	14.61
11.0	13.53	12.99	15.1	14.62	14.65
11.1	13.41	13.24	15.2	14.54	14.46
11.2	13.07	13.31	15.3	14.38	14.20
11.3	12.25	12.67	15.4	14.12	13.95
11.4	11.41	11.81	15.5	13.67	13.68
11.5	11.44	11.62	15.6	13.48	13.68
11.6	11.68	11.60	15.7	13.73	14.03
11.7	11.81	11.46	15.8	13.93	14.29
11.8	12.27	11.73	15.9	14.15	14.44
11.9	13.14	12.56	16.0	14.44	14.50
12.0	13.79	13.37			

Table B3: RCS values obtained from David Florida Laboratory for large corner reflector.

7 Appendix C: Algorithms

Function: *[scans, frequency, location, Bursthdr] = dataread(file)*

Input:	*file*	Name of file to read
Output:	*scans*	Vector containing the number of pulses in each burst
	frequency	Vector containing frequency of each burst in the file
	location	Vector containing the position in the file (in bytes) for I/Q data for each burst
	Bursthdr	String containing the time, frequency, power, pulse width, AGC, attenuation, range, transmit polarization, receive polarization, scans and burst number for each burst in the file

Algorithm:

This function reads the header information and determines the location of the raw I and Q data for each burst in the file. The function can read either binary or ASCII files.

Function: *[I, Q] = getiq(filename, location, burst_num, scans)*

Input:	*filename*	Name of input file
	location	Vector containing the position in the file (in bytes) for I/Q data for each burst
	burst_num	Number of the selected burst
	scans	Number of scans in the selected burst
Output:	*I*	In-phase data
	Q	Quadrature data

Algorithm:

This function can read either binary or ascii files. The location of the raw *I* and *Q* data in the given file is determined using the location vector and the burst number.

Function: *plotraw(I, Q, fontsize, filename, burst_num, frequency)*

Input:	*I*	In-phase data
	Q	Quadrature data
	fontsize	Size of font for displaying header information
	filename	Name of file containing data
	burst_num	Burst number (as a string)
	frequency	Frequency of the selected burst

Output: none

Global Variables:		
	CalCohTime	Calibration coherency time
	TarCohTime	Target coherency time
	Bursthdr	String containing the time, frequency, power, pulse width, AGC, attenuation, range, transmit polarization, receive polarization, scans and burst number for each burst in the file
	Izero	I-offset
	Qzero	Q-offset

Algorithm:

The I and Q offsets are first removed from the raw I and Q data using the function *convert([I Q], agc1, agc2, option, frequency)* with *option* = 1. Once the offsets are removed the coherent and non-coherent averages are computed.

The coherent average is computed by first determining the number of coherent pulses, n_c. This is determined by multiplying the pulse repetition frequency (PRF) with the either the target coherency time, if a data file is selected, or the calibration coherency time, if a calibration file is selected. The coherent average is then given by the equation

$$c_{ave} = \begin{cases} \frac{1}{q+1}\left[\sum_{k=0}^{q-1}\left|\frac{1}{n_c}\sum_{i=1}^{n_c} Z_{kn_c+i}\right| + \left|\frac{1}{r}\sum_{i=1}^{r} Z_{qn_c+i}\right|\right], & r>0 \\ \frac{1}{q}\sum_{k=0}^{q-1}\left|\frac{1}{n_c}\sum_{i=1}^{n_c} Z_{kn_c+i}\right|, & r=0 \end{cases}$$

where Z_i represents the complex sample of the ith pulse

$$Z_i = I_i + jQ_i$$

and q and r are determined such that the total number of pulses in the given burst, N, is given by

$$N = qn_c + r.$$

The non-coherent average is given by the following equation

$$n_{ave} = \frac{1}{N}\sum_{i=1}^{N}|Z_i|.$$

Both the coherent and non-coherent averages are displayed along with other header information and are not used in further calculations in this function.

When this function is executed, four graphs are displayed. The I and Q data, the phase and the amplitude are each plotted versus pulse number on separate graphs. The phase is computed using the equation

$$\varphi_i = \tan^{-1}\left(\frac{Q_i}{I_i}\right)$$

Before the phase is plotted, it is unwrapped using MATLAB's unwrap function. This function unwraps phases by changing absolute jumps greater than π to the 2π complement.

The amplitude is calculated by taking the magnitude of the complex sample of each pulse in the burst.

Function: *cal3 = proccal(cal1, cal2)*

Input:	*cal1*	matrix containing data from first calibration file
	cal2	matrix containing data from second calibration file
Output:	*cal3*	matrix containing new (combined) calibration data

Algorithm:

This function either returns a combined calibration matrix or it returns the best calibration matrix. The calibration matrices contain the following data for each burst.

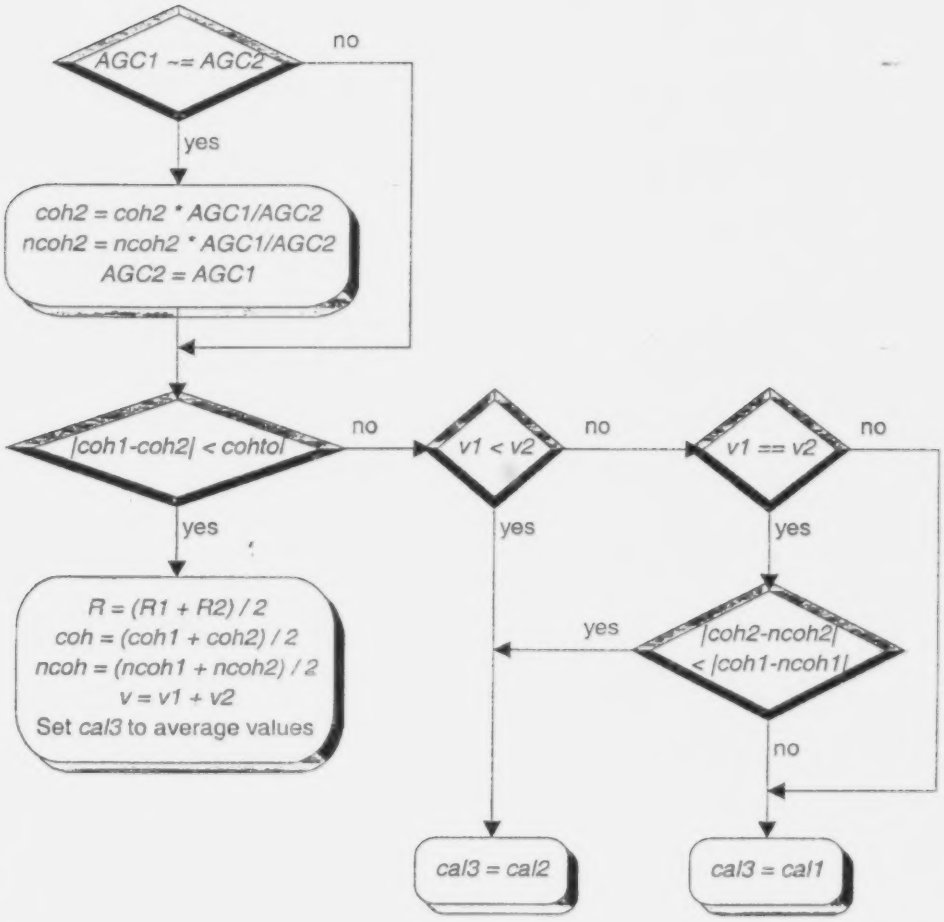
Column	Data
1	Frequency (GHz)
2	Pulse width (ns)
3	Tx polarization (H/V)
4	Rx polarization (H/V)
5	Range (m)
6	Coherent average
7	Non-coherent average
8	Calibration value (1-3)
9	AGC (power gain in dB)
10	RCS (dBsm)

Column eight of the calibration matrix is a calibration indicator value. This value is an integer, between one and three, which indicates how good the calibration data is. The higher the value, the better the calibration file. A calibration value of three indicates that a manual calibration file is used.

If the two input calibration matrices have different calibration values, then the matrix with the highest calibration value is returned. Otherwise, an averaged matrix is returned. The matrices are averaged by first finding the entries with the same frequency, pulse width and polarizations. If the AGC values from the two calibration matrices are not the same, then the coherent and non-coherent averages are adjusted so that the AGC from the first matrix is applied. Then, if the difference between the coherent averages of the two calibration matrices is less than the coherency tolerence, the range, coherent average and non-coherent average values are averaged and returned in the matrix *cal3*, otherwise the best calibration matrix is returned. The best calibration matrix is defined as the one with the highest calibration value. If the calibration values of the two files are the same, then the matrix with the smallest absolute difference between the coherent and non-coherent average is returned in matrix *cal3*.

The above algorithm is illustrated more clearly in the flow chart given below. The variables in the flow chart are defined as follows.

AGC1	Automatic gain control applied to the given burst in file 1
AGC2	Automatic gain control applied to the given burst in file 2
coh1	Coherent average of given burst from file 1
coh2	Coherent average of given burst from file 2
ncoh1	Non-coherent average of given burst from file 1
ncoh2	Non-coherent average of given burst from file 2
R1	Range to calibration device using file 1
R2	Range to calibration device using file 2
v1	Calibration value of file 1
v2	Calibration value of file 2
cohtol	Coherency tolerence (also called the coherency value in the System Configuration option)


AGC1 ~= AGC2
no
yes
coh2 = coh2 * AGC1/AGC2
ncoh2 = ncoh2 * AGC1/AGC2
AGC2 = AGC1
|coh1-coh2| < cohtol
no
yes
R = (R1 + R2) / 2
coh = (coh1 + coh2) / 2
ncoh = (ncoh1 + ncoh2) / 2
v = v1 + v2
Set cal3 to average values
v1 < v2
no
yes
v1 == v2
no
yes
|coh2-ncoh2| < |coh1-ncoh1|
yes
no
cal3 = cal2
cal3 = cal1

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

(Highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified.)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency are entered in Section 8.)

Atlantis Scientific Inc.
20 Colonnade Road, Suite 110
Nepean, Ontario, Canada, K2E 7M6

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

UNCLASSIFIED

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

DRMS Processing Software Reference Manual (U)

4. AUTHORS (Last name, first name, middle initial)

Vrzic, Sophie, S

5. DATE OF PUBLICATION (month, and year of publication of document)	6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)	6b. NO. OF REFS (total cited in document)
May, 99	55	2

7. DESCRIPTIVE NOTES (Category of the document, e.g. technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Software Reference Manual

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development, include the address.)

Defence Research Establishment Ottawa
3701 Carling Avenue, Ottawa, Ontario, K1A 0Z4

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)
	W7714-8-0153

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned to this document either by the originator or by the sponsor.)
3112	DREO CR 1999-048

11. DOCUMENT AVAILABILITY (Any limitations of further dissemination of the document, other than those imposed by security classification.)

(x) Unlimited distribution
( ) Distribution limited to defence departments and defence contractor; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractor; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.)

Full unlimited announcement

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

(Highest classification of Title, Abstract, Keywords)

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classification documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), ©, or (U). It is not necessary to include here abstracts in both official languages unless the test is bilingual).

This document describes the details of the algorithms developed for the processing software used by the DREO Radar Measurement System (DRMS). It also describes the structure of the processing software using flow charts to illustrate which programs and/or functions are executed for each of the options provided.

14. KEYWORDS< DESCRIPTIONS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. IF possible keywords should be selected from a published thesaurus. E.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

DREO Radar Measurement System
DRMS
Radar Cross Section
RCS
High Range Resolution
HRR